



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07264590 A**

(43) Date of publication of application: **13.10.95**

(51) Int. CI

H04N 7/30
H04N 1/41
H04N 5/92

(21) Application number: 06056200

(71) Applicant: NEC CORP

(22) Date of filing: 25.03.94

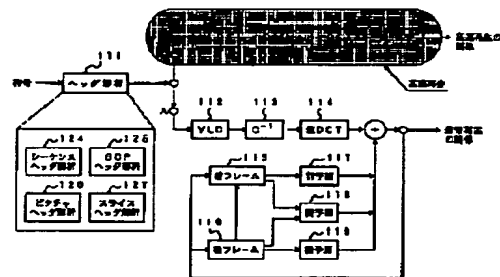
(72) Inventor: **UEDA HIROAKI**

(54) IMAGE REGENERATING DEVICE

(57) Abstract:

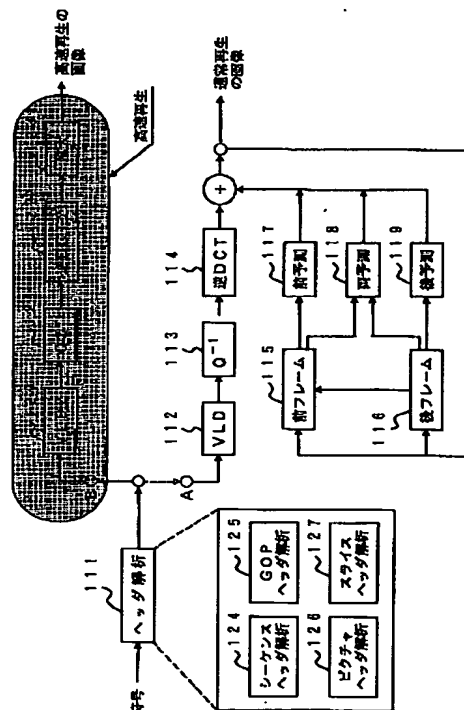
PURPOSE: To quickly regenerate a code based upon discrete cosine transformation by simple constitution without lossing picture quality by providing this picture regenerating device with a means for analyzing a code and judging whether the code fixes a parameter or not.

CONSTITUTION: In the image regenerating device, a header analysis part 111 analyzes a read code, judges whether the code fixes a parameter or not, and when the code fixes its parameter, high speed reproducing is executed. In the case of high speed regeneration, only an I picture to be an intra-frame code is regarded as a slice width fixed escape code and a highly efficiently compressed code is decoded by a VLD2 part. The decoded data are reversely quantized by a $Q2^{-1}$ part and reverse DCT is applied to the reversely quantized data by a reverse DCT2 part to display an enlarged picture. When a parameter is not fixed, normal regeneration is executed through a VLD part 112. Consequently high speed processing can be executed in accordance with the processing capacity of a regenerating device.



COPYRIGHT: (C)1995,JPO

(11)特許出願公開番号



【特許請求の範囲】

【請求項1】 画像を小ブロックに分割して、各ブロック毎に離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム内符号と、画像を小ブロックに分割して、各ブロック毎に現フレームとその前後のフレームで最も差分が小さくなるようなブロックを検索して動き補償を行い、現フレームのブロックと動き補償されたフレームのブロックで差分を取り、その差分ブロックに離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム間符号で圧縮符号化された動画を再生する画像再生装置において、符号を解析してパラメータを固定した符号であるかどうか判定する手段を備えたことを特徴とする画像再生装置。

【請求項2】 請求項1に記載した画像再生装置において、更に、表示する画像を拡大する手段を有することを特徴とする画像再生装置。

【請求項3】 請求項1に記載した画像再生装置において、更に、フレーム内符号のみに固定して伸張処理を行う手段を有することを特徴とする画像再生装置。

【請求項4】 請求項1に記載した画像再生装置において、更に、スライス幅を固定値に固定して伸張処理を行う手段を有することを特徴とする画像再生装置。

【請求項5】 請求項1に記載した画像再生装置において、更に、高周波成分を0と見なして逆離散コサイン変換の算出を行う手段を有することを特徴とする画像再生装置。

【請求項6】 請求項1に記載した画像再生装置において、更に、固定長の符号のみに固定して復号する手段を有することを特徴とする画像再生装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、離散コサイン変換(DCT、以下DCTという)をベースとした符号化方式(JPEGやMPEGなど)で圧縮符号化された画像符号を再生する装置に関する。

【0002】

【従来の技術】 画像をデジタル化してCD-ROMやハードディスクなどの記録媒体に記録する場合、そのデータ量は巨大なものとなるため通常は圧縮符号化して記録される。

【0003】 圧縮符号化方式の中で使用されることが多いDCTをベースとした符号化方式があるが、これはJPEG(Joint Photographic Expert Group)やMPEG(Moving Pictures Expert Group)などの国際標準である符号化方式で採用されている。

【0004】 従来のDCTをベースとした符号化方式による画像符号の再生についてMPEGを例として図面を参照して説明する。図16はMPEGに準拠した画像符号の再生を説明する図である。図16に示すように符号

を読み込んで、ヘッダ解析201で符号の種類などを解析する。MPEGではフレーム内符号であるIピクチャと、前方向のみのフレーム間符号であるPピクチャと、前後の双方向のフレーム間符号であるBピクチャの3種類に分かれている。

【0005】 Iピクチャの場合はVLD202で高能率圧縮された可変長のハフマン符号を復号して、Q⁻¹203で逆量子化して、IDCT204で逆DCT処理によりブロックの画素の値を算出して、画像を伸張する。

10 【0006】 また、Pピクチャの場合はVLD202で復号して、Q⁻¹203で逆量子化して、IDCT204で逆DCT処理によりブロックの差分値を算出して、前予測207により前フレーム205に格納された前フレームの動き補償したブロックに差分値を加算して、画像を伸張する。

【0007】 また、Bピクチャの場合はVLD202で復号して、Q⁻¹203で逆量子化して、IDCT204で逆DCT処理によりブロックの差分値を算出して、両予測208または後予測209により前フレーム205に格納された前フレームの動き補償したブロックと後フレーム206に格納された後フレームの動き補償したブロックに差分値を加算して、画像を伸張する。

【0008】 このように国際標準であるMPEGに基づいた再生機であればどの装置でもMPEGの符号を再生することができる。しかし、逆DCTやVLDなどの処理にはCPUの負担が大きいので、高速なCPUでなければ高速再生はできない。例えば、JPEGやMPEGなどの再生機で15フレーム/秒の処理を行うには1フレームの再生処理は約66ミリ秒で行う必要がある。もし、ハフマン符号を復号するのに30ミリ秒かかり、逆量子化するのに10ミリ秒かかり、逆DCTに20ミリ秒かかり、表示に20ミリ秒かかると全体の処理時間が80ミリ秒かかるので、1フレームの画像を再生するのに14ミリ秒遅れる。

【0009】 そこで、CPUの負担を少なくするために符号化方式の一部を変えてDCT計算の代わりに近傍画素との差分計算にしたり、量子化を省いたり、可変長の符号の代わりに4ビット単位の符号で符号化することなどによって、低速なCPUでも高速に再生できるようにすることが考えられる。

40 【0010】 この従来例として特開昭63-95791があるが、この方式ではブロックのデータの長さが等しくなるように可変長データを分配することにより、CPUに負担をかけずに再生できるようにしている。

【0011】 また、特開平4-56492では1ブロックのDCT変換後の係数が全て0である場合には無効ブロックとして、無効ブロックを示す情報のみを別信号で送ることにより、処理量を削減している。

【0012】 また、CPUの負担を少なくするために、符号化するときに圧縮符号量を制御する方法も考えられ

る。その従来例として、特開平4-329089があるが、この方式ではフレーム内で小領域に分割されたブロック毎の情報量に応じて符号量制御を行い、各ブロック毎に割り当てる符号量とステップサイズを最適な状態に設定することで、符号量を制御している。

【0013】

【発明が解決しようとする課題】しかしながら上記のような符号化方式の一部を変える画像圧縮では国際標準の符号化方式と互換性が取れなくなるために、専用の再生機が必要となる問題点がある。

【0014】また、圧縮符号量の制御を行う画像圧縮では複数回の圧縮を繰り返す必要があるため、圧縮符号を作成するのに時間がかかる。また、再生機の処理能力に応じた符号量制御ではないため、処理速度の違い再生機ではリアルタイムの再生ができない。また、圧縮符号量が決まっているので高速な再生機でも決められた画質の符号しか再生できないという問題点がある。

【0015】そこで、本発明の目的は再生機の能力に応じて画質をできるだけ損なわずに、簡単な構成によって、DCTをベースとした符号を高速に再生できるようにした画像再生装置や画像再生方式を提供することにある。

【0016】

【課題を解決するための手段】本発明によれば、画像を小ブロックに分割して、各ブロック毎に離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム内符号と、画像を小ブロックに分割して、各ブロック毎に現フレームとその前後のフレームで最も差分が小さくなるようなブロックを検索して動き補償を行い、現フレームのブロックと動き補償されたフレームのブロックで差分を取り、その差分ブロックに離散コサイン変換を行い、該変換結果を量子化して、高能率符号化されたフレーム間符号で圧縮符号化された動画像を再生する画像再生装置において、符号を解析してパラメータを固定した符号であるかどうか判定する手段を備えたことを特徴とする画像再生装置が得られる。

【0017】また、本発明によれば、更に、表示する画像を拡大する手段を有することを特徴とする画像再生装置が得られる。

【0018】また、本発明によれば、更に、フレーム内符号のみに固定して伸張処理を行う手段を有することを特徴とする画像再生装置が得られる。

【0019】また、本発明によれば、更に、スライス幅を固定値に固定して伸張処理を行う手段を有することを特徴とする画像再生装置が得られる。

【0020】また、本発明によれば、更に、高周波成分を0と見なして逆離散コサイン変換の算出を行う手段を有することを特徴とする画像再生装置が得られる。

【0021】また、本発明によれば、更に、固定長の符号のみに固定して復号する手段を有することを特徴とす

る画像再生装置が得られる。

【0022】更に、本発明によれば、上記各手段を組み合わせた画像再生装置を得ることもできる。

【0023】

【作用】本発明によれば、再生機の処理能力に応じて画質をできるだけ損なわずに、高速に再生できる。また、国際標準である符号化方式と互換性が取れるので、専用の再生機を必要としない。

【0024】

- 10 【実施例】次に本発明についてMPEGを例として図面を参照して説明する。図1は本発明の一実施例を示す画像再生装置のブロック図である。図1の画像再生装置はヘッダ解析111とVLD112とQ⁻¹113と逆DCT114と前フレーム115と後フレーム116と前予測117と両予測118と後予測119から構成される。また、ヘッダ解析111はシーケンスヘッダ解析124とGOPヘッダ解析125とピクチャヘッダ解析126とスライスヘッダ解析127から構成される。画像再生装置は符号を読み込んでヘッダ解析111を行う。
- 20 その時、パラメータを固定した符号であるかどうか判断して、そうである場合は高速再生を行う。高速再生の場合にはIピクチャのみ、スライス幅固定、エスケープ符号のみと見なして、VLD2(120)で高能率圧縮された符号を復号して、Q⁻¹121で逆量子化して、逆DCT2(122)で逆DCTを行って、拡大123で再生する画像を拡大して表示する。高速再生でない場合には、フレーム内符号であるIピクチャと、前方向のみのフレーム間符号であるPピクチャと、前後の双方向のフレーム間符号であるBピクチャの3種類の符号を伸張する。
- 30

【0025】Iピクチャの場合はVLD112で復号して、Q⁻¹113で逆量子化して、逆DCT114で逆DCT処理によりブロックの画素の値を算出して、画像を伸張する。

【0026】また、Pピクチャの場合はVLD112で復号して、Q⁻¹113で逆量子化して、逆DCT114で逆DCT処理によりブロックの差分値を算出して、前予測117により前フレーム115に格納された前フレームの動き補償したブロックに差分値を加算して、画像を伸張する。

40 【0027】また、Bピクチャの場合はVLD112で復号して、Q⁻¹113で逆量子化して、逆DCT114で逆DCT処理によりブロックの差分値を算出して、両予測118または後予測119により前フレーム115に格納された前フレームの動き補償したブロックと後フレーム116に格納された後フレームの動き補償したブロックに差分値を加算して、画像を伸張する。

50 【0028】また、図2にMPEGに準拠した符号フォーマットの階層図を示す。MPEGの符号は図2に示すようにいくつかの階層構造となっている。一番上の階層

がビデオ・シーケンスであり、複数のGOP (Group Of Picture) から構成される。GOPは複数のピクチャから構成され、1つのピクチャが1枚の画像を示している。ピクチャは任意の領域に分割された複数のスライスから構成される。スライスは左から右へ、または上から下への順序で並んだ複数のマクロブロックから構成される。マイクロブロックは16×16ドットのブロックを更に8×8ドットのブロックに分割した輝度成分(Y1, Y2, Y3, Y4)と輝度成分に一致する領域の8×8ドットのブロックの色差成分(Cb, Cr)の6個のブロックから構成される。8×8ドットのブロックが符号化の最小単位となる。

【0029】また、図3にMPEGに準拠した符号フォーマットの構成図を示す。MPEGの符号は図3に示すように各階層毎に(1)シーケンスヘッダと、(2)GOP (Group Of Picture) ヘッダと、(3)ピクチャヘッダと、(4)スライスヘッダと、(5)マクロブロックヘッダと、(6)ブロックの符号とから構成される。

【0030】シーケンスヘッダにはUserData (11)に示すようにユーザが自由に定義できるユーザデータの領域があり、その領域にパラメータを固定したことを示すフラグを格納する。そのユーザデータ(11)は固定パラメータで圧縮した符号であることを示す識別子“Custom”(12)と、固定パラメータのフラグ(13)で構成される。固定パラメータのフラグ(13)は伸張時に拡大することを示すビットと、Iピクチャのみに固定することを示すビットと、スライス幅はピクチャサイズに固定することを示すビットと、高周波成分をカットすることを示すビットと、エスケープ符号のみに固定することを示すビットで構成される。

【0031】次に、以上の構成を有する本実施例の画像再生装置における画像再生処理について説明する。

【0032】図4は画像再生装置の拡大を説明する図である。図4では2×2画素の画像を縦2倍、横2倍に拡大する場合の例を示している。図4の(A)は2×2画素の各画素の値を示し、(C)は縦2倍、横2倍に拡大後の4×4画素の各画素の値を示している。また、図4の(B)は1画素を4つの画素に変換する処理を示している。図4の例では(A)から1画素取り出して、それを(B)のように4つの画素に変換して、縦2倍、横2倍に拡大した画像(C)に書き込んでいる。例えば、

(A)の右上の画素(29)は(B)の変換により、P1(29)とP2(29)とP3(29)とP4(29)となり、(C)の対応する領域に書き込まれる。

【0033】図4の例では縦2倍、横2倍に拡大するときに画質を向上するために画素の値の補正(ディサ処理)を行っていないが、ディサ処理を行っても良い。この場合例えば、図4の(B)で $P1 = P0 + a$ 、 $P2 =$

$P0 + b$ 、 $P3 = P0 + c$ 、 $P4 = P0 + d$ (a, b, c, dは任意の整数)として、4倍拡大を行う。また、倍率は縦2倍、横2倍としているが、3以上の整数倍でも良い。

【0034】また、図5は図4の動作を説明するフローチャートである。図5に示すように拡大は拡大前の画像から画素を1つ読み込んで(ステップ1)、読み込んだ1つの画素(P0)を4つの画素($P1 = P0$ 、 $P2 = P0$ 、 $P3 = P0$ 、 $P4 = P0$)に変換して(ステップ2)、変換した4つの画素を拡大後の画像に書き込む(ステップ3)。次に全画素を処理したかどうか判断して(ステップ4)、そうでない場合はステップ1に戻り、そうでない場合は処理を終了する。

【0035】このように伸張した画像を拡大して表示することにより、伸張時間を短縮できる。例えば、縦2倍、横2倍に拡大する場合は伸張時間は1/4に短縮できる。

【0036】また、拡大の大部分の処理はメモリのリード/ライトである。伸張の処理にはメモリのリード/ライトだけでなく、逆DCTや逆量子化での乗算や可変長符号の復号でのビットシフトの処理がある。メモリのリード/ライトの方が乗算やビットシフトよりもCPUの負担が小さいので、拡大処理を含めても、拡大しない場合よりも拡大した場合の方が高速に処理できる。

【0037】図6はピクチャ構成を固定にした場合の処理を説明する図である。MPEGではIピクチャ(フレーム内符号)とPピクチャ(前方向予測符号)とBピクチャ(双方向予測符号)の3つのピクチャがあるが、図6の(A)はIピクチャとPピクチャとBピクチャから成るピクチャ構成を示し、(B)はIピクチャのみに固定したピクチャ構成を示している。Pピクチャの場合は前ピクチャの画像を参照して前予測を行う必要があり、またBピクチャの場合は前フレームの画像や後フレームの画像を参照して後予測や両予測を行う必要があるので、Iピクチャの場合と比べて処理時間がかかる。図6の(A)の例では1フレーム目はIピクチャであり参照フレームは無い。2フレーム目と2フレーム目はBピクチャであり、前フレームである1フレーム目の画像と後フレームである4フレーム目の画像を参照して、後予測や両予測を行う。4フレームはPピクチャであり、前フレームである1フレーム目の画像を参照して前予測を行う。図6の(B)の例では1フレーム目から4フレーム目まで全てIピクチャなので、前予測や後予測や両予測を行う必要は無い。

【0038】また、図7は図6の動作を説明するフローチャートである。図7の(A)はIピクチャとPピクチャとBピクチャから成るピクチャ構成を処理する場合を示し、(B)はIピクチャのみに固定したピクチャ構成を処理する場合を示し、(C)は(A)と(B)のどちらを処理するか判断する処理を示している。図7の

(A) に示すピクチャ処理1はピクチャヘッダからピクチャの種別を読み込んで(ステップ10)、ピクチャの種別を判断して(ステップ11)、Iピクチャの場合はIピクチャの処理を行う(ステップ12)。Pピクチャの場合はPピクチャの処理を行う(ステップ13)。Bピクチャの場合はBピクチャの処理を行う(ステップ13)。図7の(B)に示すピクチャ処理2はIピクチャの処理を行う(ステップ15)。図7の(C)に示すピクチャヘッダ解析はパラメータ固定であるかどうか判断して(ステップ16)、そうでない場合はピクチャ処理1を行う(ステップ17)。そうである場合はピクチャ処理2を行う(ステップ18)。

【0039】このようにピクチャ構成をIピクチャのみに固定することにより、ピクチャヘッダからピクチャ種別を読み込んで判断する処理を省略でき、Pピクチャの処理の前予測やBピクチャの処理の後予測や両予測を省略できるので、高速に処理できる。

【0040】図8はスライス幅を固定した場合の処理を説明する図である。MPEGでは1フレームの画像を16×16画素を単位としていくつかの領域に分割している。分割した領域の符号の先頭にはスライスヘッダが挿入されて、画像のどの領域であるかを示している。図8の(A)はスライス幅を固定していない場合を示し、図8の(B)はスライス幅をピクチャサイズに固定した場合を示している。図8の(A)は画像を5つの領域(1～5)に分割している。1から5までの各領域の符号の先頭にはスライスヘッダが挿入される。図8の(B)ではスライス幅をピクチャサイズに固定しているので、領域を分割していない。そのため、スライスヘッダも1つしかない。

【0041】また、図9は図8の動作を説明するフローチャートである。図9の(A)はスライス幅を固定していない場合の処理を示し、(B)はスライス幅をピクチャサイズに固定した場合の処理を示し、(C)は(A)と(B)のどちらを処理するか判断する処理を示している。図9の(A)に示すスライス処理1はスライスヘッダを読み込んで(ステップ20)、マクロブロックの処理を行い(ステップ21)、次のヘッダがスライスヘッダであるかどうか判断して(ステップ22)、そうである場合はステップ20へ戻る。そうでない場合は全マクロブロックの処理が終了したかどうか判断して(ステップ23)、そうでない場合はステップ21へ戻り、そうである場合は処理を終了する。図9の(B)に示すスライス処理2はスライスヘッダを読み込んで(ステップ24)、マクロブロックの処理を行い(ステップ25)、全マクロブロックの処理が終了したかどうか判断して(ステップ26)、そうでない場合はステップ24へ戻り、そうである場合は処理を終了する。図9の(C)に示すスライスヘッダ解析はパラメータ固定であるかどうか判断して(ステップ27)、そうでない場合はスライ

ス処理1を行う(ステップ28)。そうである場合はスライス処理2を行う(ステップ29)。

【0042】このようにスライス幅をピクチャサイズに固定することにより、スライスヘッダを読み込んで解析する処理回数を削減できるので、高速に処理できる。

【0043】図10は高周波成分を0と見なして逆DCTを行う場合の処理を説明する図である。図10の

(A)は高周波成分を0と見なさない場合を示し、図10の(B)は高周波成分を0と見なす場合を示している。

図10の(A)は8×8ブロックの周波数成分の低周波から高周波へのスキャン順序(ジグザグスキャン)を示している。図10の(B)もジグザグスキャンを示しているが、22番目以降の高周波成分を全て0と見なしている。自然画像の場合は画像情報が低周波数成分に集中するので、高周波数成分を0と見なしても、画質は大きく劣化して見えない。

【0044】また、図11は高周波成分を0と見なせずに逆DCTを行う場合の動作を説明するフローチャートである。図11に示す逆DCTは変数yに0を格納して(ステップ30)、変数vに0を格納して(ステップ31)、変数ddに0を格納して(ステップ32)、変数uに0を格納して(ステップ33)。次に変数ddに逆DCTを行う8×8ブロックのバッファのBuffer(y, u)と逆DCTの係数である $\cos((2v+1)u\pi)/2$ の積を加算する(ステップ34)。次に変数uに1を加算して(ステップ35)、変数uの値が8より小さいかどうか判断して(ステップ36)、そうである場合はステップ34へ戻る。そうでない場合は一時的な格納バッファのt(v, y)に変数ddの値を格納する(ステップ37)。次に変数vに1を加算して(ステップ38)、変数vの値が8より小さいかどうか判断して(ステップ39)、そうである場合はステップ32へ戻る。次に変数yに1を加算して(ステップ40)、変数yの値が8より小さいかどうか判断して(ステップ41)、そうである場合はステップ31へ戻る。そうでない場合は変数yに0を格納する(ステップ42)。次に変数xに0を格納して(ステップ43)、変数ddに0を格納して(ステップ44)、変数uに0を格納する(ステップ45)。次に変数ddにt(y, u)と $\cos((2x+1)u\pi)/2$ の積を加算する(ステップ46)。次に変数uに1を加算して(ステップ47)、変数uの値が8より小さいかどうか判断して(ステップ48)、そうである場合はステップ46へ戻る。そうでない場合はBuffer(x, y)に変数ddの値を格納する(ステップ49)、次に変数xに1を加算して(ステップ50)、変数xの値が8より小さいかどうか判断して(ステップ51)、そうである場合はステップ44へ戻る。次に変数yに1を加算して(ステップ52)、変数yの値が8より小さいかどうか判断して(ステップ53)、そうである場合はステップ44へ

戻る。そうでない場合は処理を終了する。

【0045】また、図12は高周波成分を0と見なし逆DCTを行う場合の動作を説明するフローチャートである。図12は22番目以降の周波数成分を0と見なした例である。図12に示す逆DCT2は変数yに0を格納して(ステップ60)。変数vに0を格納して(ステップ61)、変数ddに0を格納して(ステップ62)、変数uに0を格納する(ステップ63)。次に変数ddに逆DCTを行う8×8ブロックのバッファのBuffer(y, u)と逆DCTの係数である $\cos((2v+1)u\pi)/2$ の積を加算する(ステップ64)。次に変数uに1を加算して(ステップ65)、変数uの値がumax(y)の値より小さいかどうか判断して(ステップ66)、そうである場合はステップ64へ戻る。そうでない場合は一時的な格納バッファのt(v, y)に変数ddの値を格納する(ステップ67)。次に変数vに1を加算して(ステップ68)、変数vの値が8より小さいかどうか判断して(ステップ69)、そうである場合はステップ62へ戻る。次に変数yに1を加算して(ステップ70)、変数yの値が6より小さいかどうか判断して(ステップ71)、そうである場合はステップ61へ戻る。そうでない場合は変数yに0を格納する(ステップ72)。次に変数xに0を格納して(ステップ73)、変数ddに0を格納して(ステップ74)、変数uに0を格納する(ステップ75)。次に変数ddにt(y, u)と $\cos((2x+1)u\pi)/2$ の積を加算する(ステップ76)。次に変数uに1を加算して(ステップ77)、変数uの値が6より小さいかどうか判断して(ステップ78)、そうである場合はステップ76へ戻る。そうでない場合はBuffer(x, y)に変数ddの値を格納する(ステップ79)。次に変数xに1を加算して(ステップ80)、変数xの値が8より小さいかどうか判断して(ステップ81)、そうである場合はステップ74へ戻る。次に変数yに1を加算して(ステップ82)、変数yの値が8より小さいかどうか判断して(ステップ83)、そうである場合はステップ74へ戻る。そうでない場合は処理を終了する。

【0046】図12の例では22番目以降を0と見なしているが、再生機の処理能力に応じて1～64の値でも良い。

【0047】このように高周波成分を0と見なさない場合の乗算回数は $8 \times 8 \times 8 \times 2 = 1024$ 回であり、22番目以降の周波数成分を0と見なした場合の乗算回数は $8 \times (6 + 5 + 4 + 3 + 2 + 1) \times 6 + 6 \times 8 \times 8 = 510$ 回であるので、高周波成分を0と見なし逆DCTを行った方が高速に処理できる。

【0048】図13はエスケープ符号の構成図である。MPEGには固定ピッチ長のエスケープ符号があり、それは(1)エスケープ符号であることを示す符号(00

0001)と(2)ラン長(無効係数の数)と(3)レベル(有効係数の値: -128～+128)から構成される。

【0049】図14は通常の符号とエスケープ符号の例を示した表である。図14に示すように通常の符号は2～17ビットの可変長の符号であるが、エスケープ符号は全て20ビットの固定長の符号となる。

【0050】また、図15はVLDの動作を説明するフローチャートである。図15の(A)は通常の符号の場合の処理を示し、図15の(B)はエスケープ符号のみに固定した場合の処理を示している。図15の(A)に示すVLDは符号を8ビット取り出し(ステップ90)、8ビットの値から符号の種類を判断して(ステップ91)、8ビット以下の符号であるかどうかを判断して(ステップ92)、そうでない場合はステップ94へ進む。そうでない場合はさらに必要なビット数分取り出して(ステップ93)、符号のビット数を符号バッファのポインタに加算して(ステップ94)、符号に対応するラン長とレベルを取り出して(ステップ95)、ラン長とレベルを返す(ステップ96)。図15の(B)に示すVLD2は符号を8ビット取り出し(ステップ97)、bit0～bit7からレベルを取り出し(ステップ98)、bit8～bit13からラン長を取り出し(ステップ99)、20ビットを符号バッファのポインタに加算して(ステップ100)、ラン長とレベルを返す(ステップ101)。

【0051】このように通常の符号の場合は符号ビット数を調べて、その値に応じて処理しなければならないが、エスケープ符号のみの場合は20ビットの固定長なので、符号ビット数を調べる必要がなく、高速に処理できる。

【0052】

【発明の効果】以上説明したように本発明は、パラメータを固定して伸張することができるので、再生機の処理野兎力に応じて画質をできるだけ損わずに、高速に再生できる。また、国際標準である符号化方式と互換性が取れるので、専用の再生機を必要としない。

【図面の簡単な説明】

【図1】本発明の一実施例の動画像再生装置のフロー図である。

【図2】MPEGの符号の階層図である。

【図3】MPEGの符号の構成図である。

【図4】本発明の拡大処理を説明する図である。

【図5】本発明の拡大処理の動作を示すフローチャートである。

【図6】本発明のピクチャ構成を固定した処理を説明する図である。

【図7】本発明のピクチャ処理の動作を示すフローチャートである。

【図8】本発明のスライス幅を固定した処理を説明する

図である。

【図9】本発明のスライス処理の動作を示すフローチャートである。

【図10】本発明の高周波成分を0と見なす処理を説明する図である。

【図11】逆DCTの動作を示すフローチャートである。

【図12】本発明の逆DCTの動作を示すフローチャートである。

【図13】エスケープ符号の構成図である。

【図14】通常の符号とエスケープ符号の例を示した表である。

【図15】本発明のVLDの動作を示すフローチャートである。

【図16】従来例の動画再生装置のフロー図である。

【符号の説明】

111 ヘッダ解析

*

* 112

113

114

115

116

117

118

119

120

10 121

122

123

124

125

126

127

VLD

Q^{-1}

逆DCT

前フレーム

後フレーム

前予測

両予測

後予測

VLD2

$Q2^{-1}$

逆DCT2

拡大

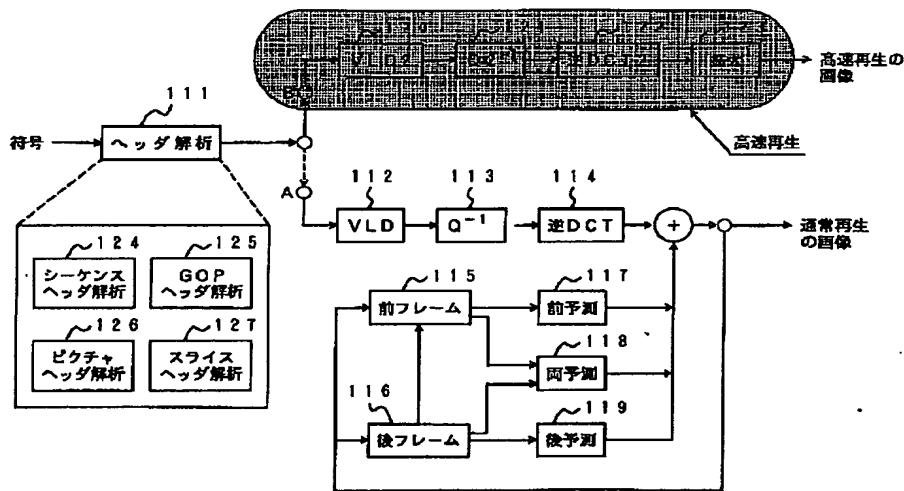
シーケンスヘッダ解析

GOPヘッダ解析

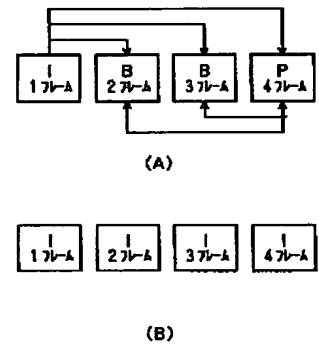
ピクチャヘッダ解析

スライスヘッダ解析

【図1】

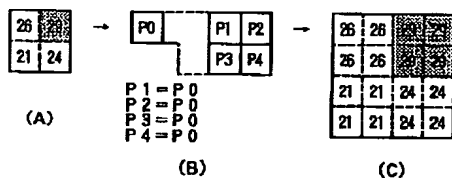


【図6】

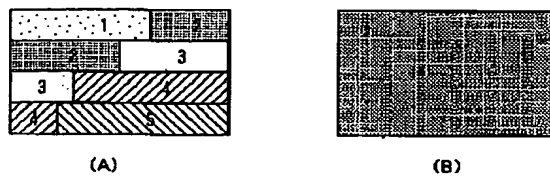


I: フレーム内符号
P: 前方予測符号
B: 後方予測符号

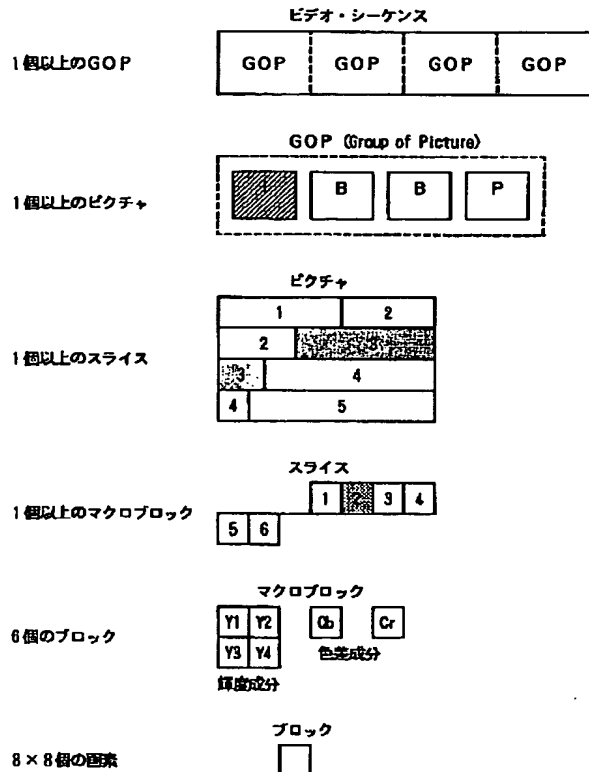
【図4】



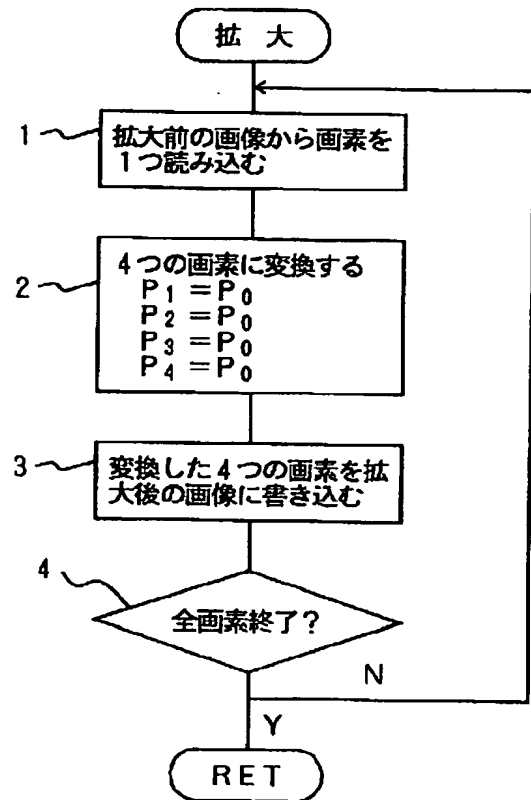
【図8】



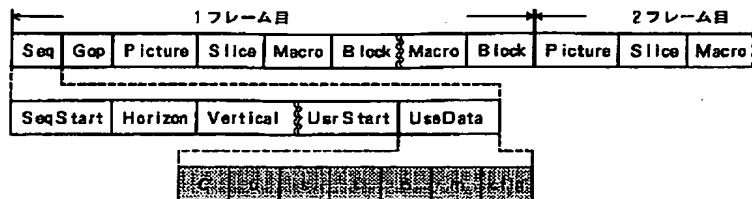
【図2】



【図5】



【図3】



- (1) Seq : シーケンスヘッダ
 (2) Gop : GOP (Group of Picture) ヘッダ
 (3) Picture : ピクチャヘッダ
 (4) Slice : スライスヘッダ
 (5) Macro : マクロブロックヘッダ
 (6) Block : ブロックの符号
 (7) SeqStart : シーケンスヘッダのスタートコード
 (8) Horizon : 横方向の画素数
 (9) Vertical : 縦方向の画素数
 (10) UserStart : ユーザデータの存在を示すコード
 (11) UserData : ユーザデータ
 (12) "Custom" : 固定パラメータで圧縮した符号であることを示す識別子
 (13) cfig : 固定パラメータのフラグ
- MSB 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 LSB

- ① B 0 = 1 の場合 : 伸張時に拡大する
 ② B 1 = 1 の場合 : 1 ピクチャのみに限定する
 ③ B 2 = 1 の場合 : スライス幅はピクチャサイズに固定する
 ④ B 3 = 1 の場合 : 高周波成分をカットする
 ⑤ B 4 = 1 の場合 : エスケープ符号のみに限定する

【図10】

1	2	6	7	15	16	28	29	1	2	6	7	15	16	x	x
3	5	8	14	17	27	30	43	3	5	8	14	17	x	x	x
4	9	13	18	26	31	42	44	4	9	13	18	x	x	x	x
10	12	19	25	32	41	45	54	10	12	19	x	x	x	x	x
11	20	24	33	40	46	53	55	11	20	x	x	x	x	x	x
21	23	34	39	47	52	56	61	21	x	x	x	x	x	x	x
22	35	38	48	51	57	60	62	x	x	x	x	x	x	x	x
36	37	49	50	58	59	63	64	x	x	x	x	x	x	x	x

(A)

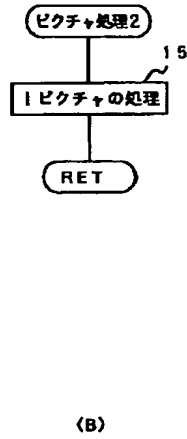
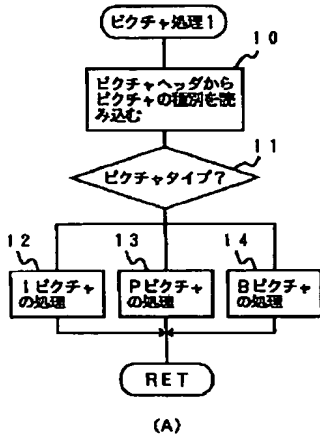
(B)

xの箇所は0と見なして計算する

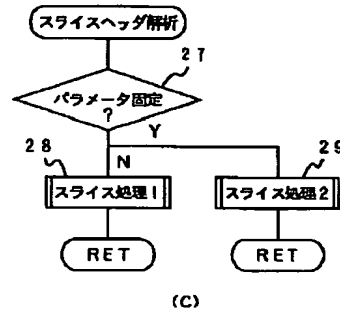
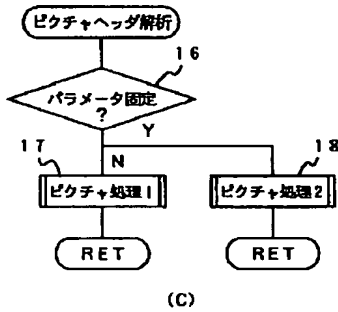
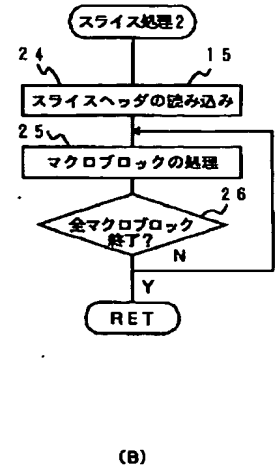
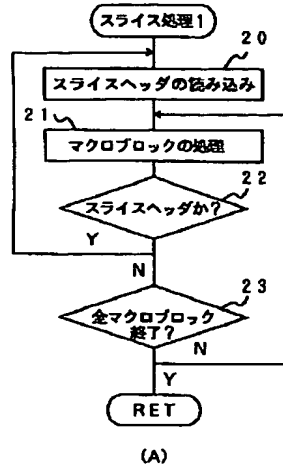
【図14】

ラン	レベル	通常符号 2~17ビット	エスケープ符号 20ビット固定
2	1	0 1010	0000 0100 0010 0000 0001
1	4	000 0001 1000	0000 0100 0100 0000 0100
2	5	00 0000 0010 1000	0000 0100 0101 0000 0101

【図 7】



【図 9】

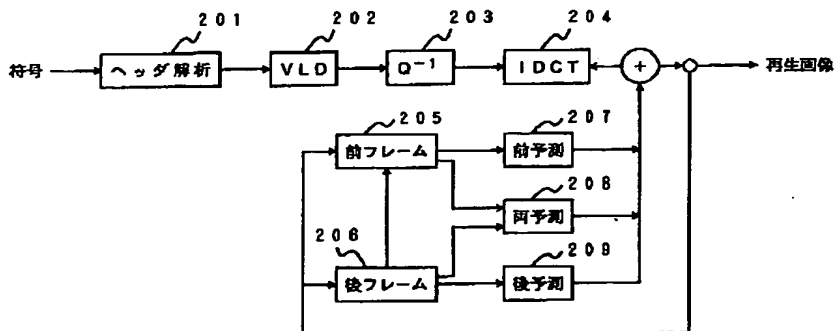


【図 13】

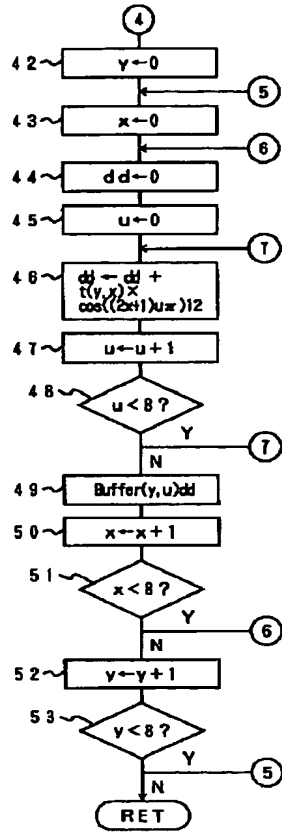
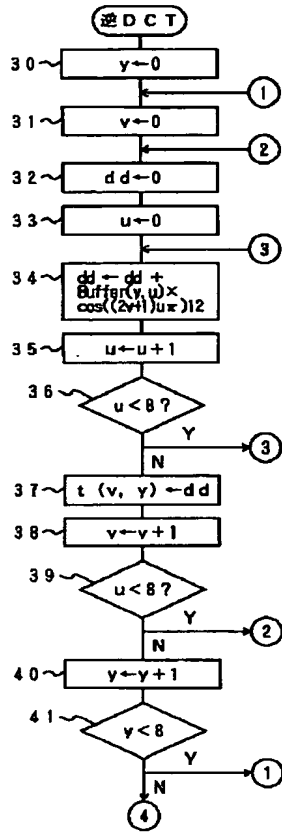
Esc						Run						Level							
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x

- (1) Esc : エスケープ符号であることを示す符号
 (2) Run : ラン長 (有効ビットの数)
 (3) Level : レベル (有効ビットの値: -128 ~ +128)

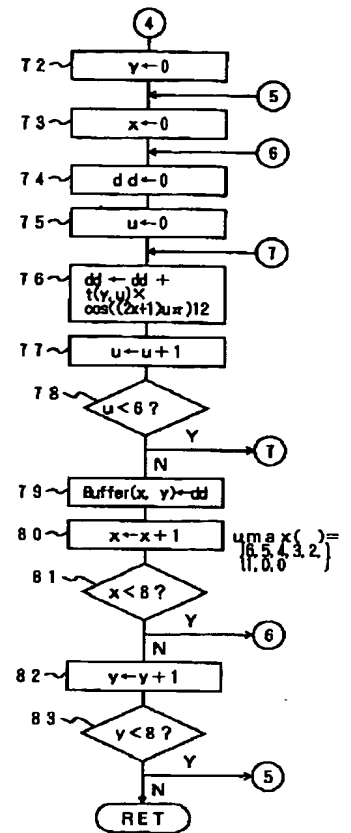
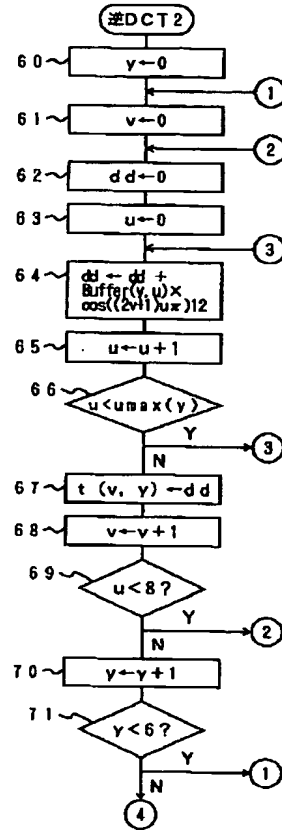
【図 16】



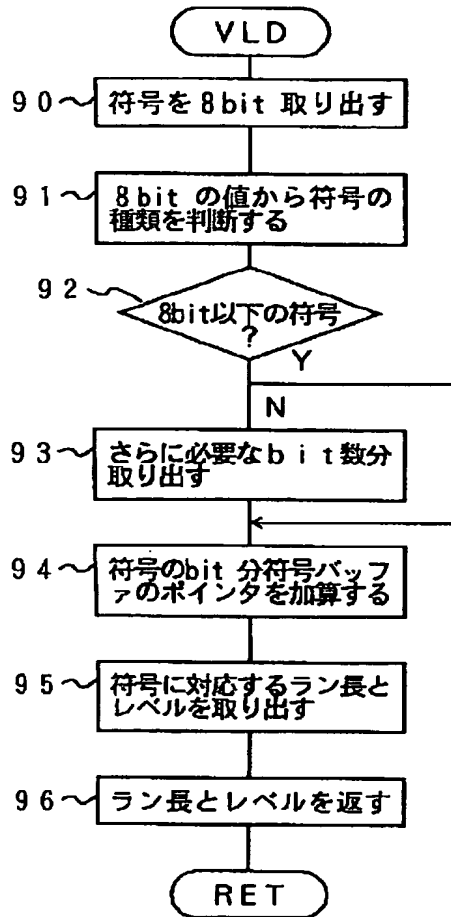
【図11】



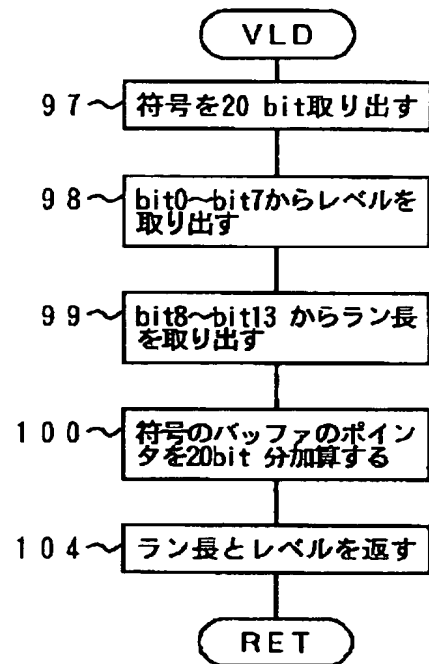
【図12】



【図15】



(A)



(B)